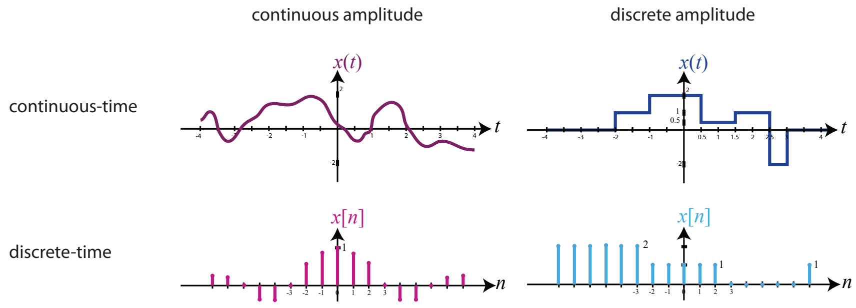
Introduction to DSP Systems

Dr. Deepa Kundur

University of Toronto

Analog and Digital Signals

- analog signal = continuous-time + continuous amplitude
- digital signal = discrete-time + discrete amplitude



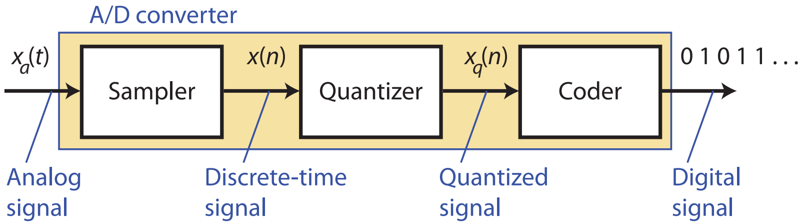
Analog and Digital Signals

- Analog signals are fundamentally significant because we must interface with the real world which is analog by nature.
- Digital signals are important because they facilitate the use of digital signal processing (DSP) systems, which have practical and performance advantages for several applications.

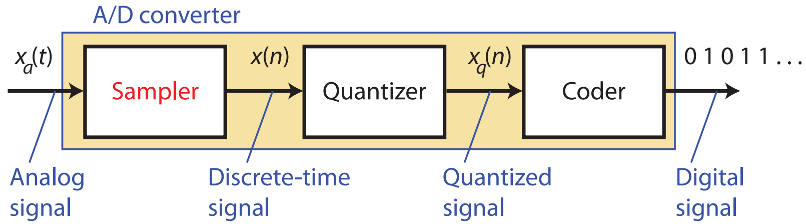
Analog and Digital Systems

- analog system =
  analog signal input + analog signal output
  - advantages: easy to interface to real world, do not need A/D or D/A converters, speed not dependent on clock rate
- digital system =
  digital signal input + digital signal output
  - advantages: re-configurability using software, greater control over accuracy/resolution, predictable and reproducible behavior

Analog-to-Digital Conversion



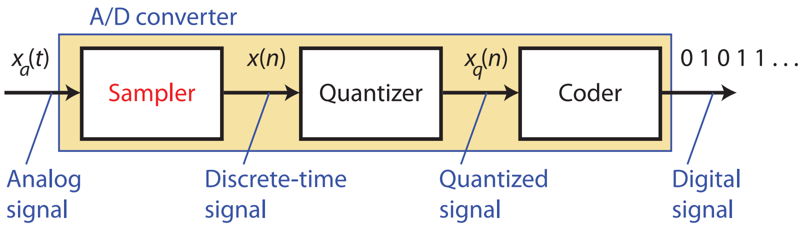
Analog-to-Digital Conversion



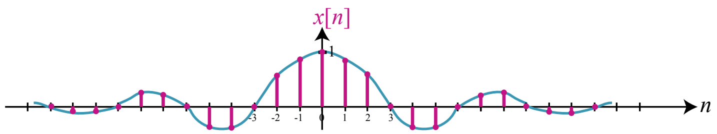
Sampling:

- conversion from cts-time to dst-time by taking “samples” at discrete time instants
- E.g., uniform sampling: $x(n) = x_a(nT)$ where T is the sampling period and $n \in \mathbb{Z}$

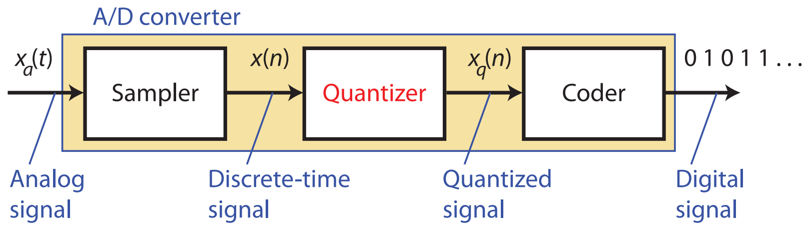
Analog-to-Digital Conversion



Sampling:



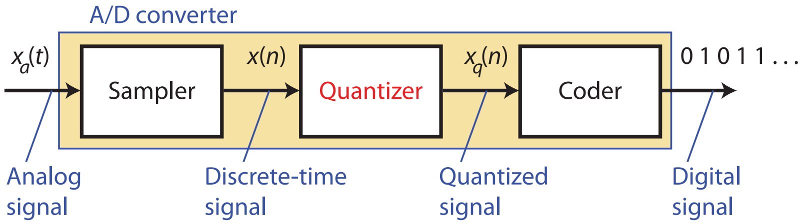
Analog-to-Digital Conversion



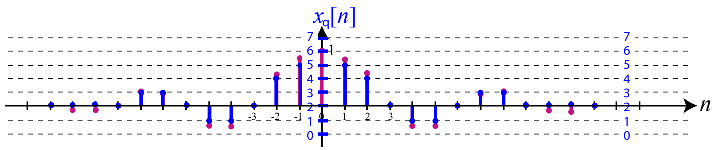
Quantization:

- conversion from dst-time cts-valued signal to a dst-time dst-valued signal
- quantization error: $e_q(n) = x_q(n) - x(n)$ for all $n \in \mathbb{Z}$

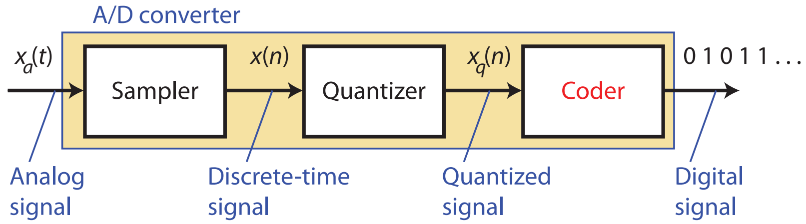
Analog-to-Digital Conversion



Quantization:



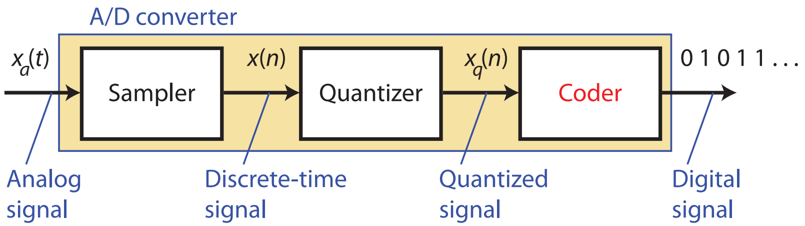
Analog-to-Digital Conversion



Coding:

- representation of each dst-value $x_q(n)$ by a *b*-bit binary sequence
- e.g., if for any n, $x_q(n) \in \{0, 1, \ldots, 6, 7\}$, then the coder may use the following mapping to code the quantized amplitude:

Analog-to-Digital Conversion



Example coder:

0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Sampling Theorem

If the highest frequency contained in an analog signal $x_a(t)$ is $F_{max} = B$ and the signal is sampled at a rate

$$F_s > 2F_{max} = 2B$$

then $x_a(t)$ can be exactly recovered from its sample values using the interpolation function

$$g(t) = \frac{\sin(2\pi Bt)}{2\pi Bt}$$

Note: $F_N = 2B = 2F_{max}$ is called the Nyquist rate.

Sampling Theorem

$$\text{Sampling Period} = T = \frac{1}{F_s} = \frac{1}{\text{Sampling Frequency}}$$

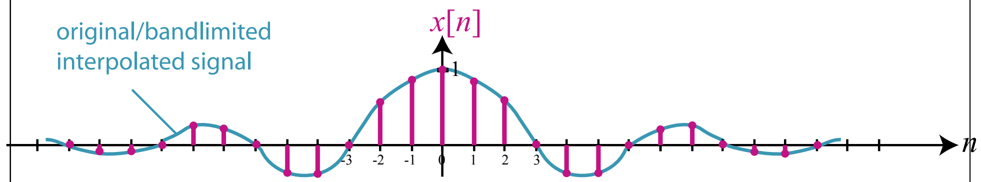
Therefore, given the interpolation relation, $x_a(t)$ can be written as

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT) g(t - nT)$$

$$x_a(t) = \sum_{n=-\infty}^{\infty} x(n) \; g(t - nT)$$

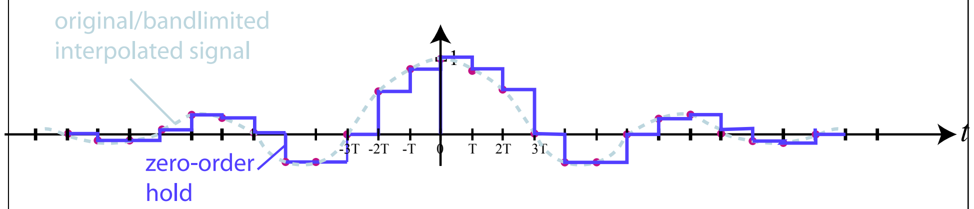
where $x_a(nT) = x(n)$; called bandlimited interpolation.

Digital-to-Analog Conversion



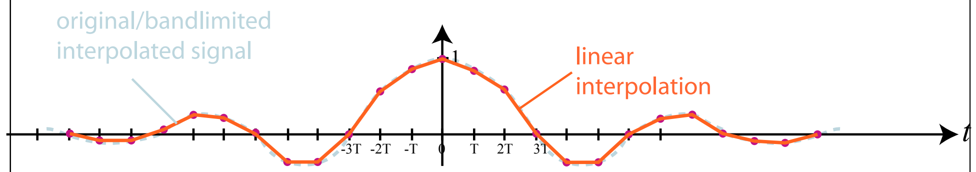
- Common interpolation approaches: bandlimited interpolation, zero-order hold, linear interpolation, higher-order interpolation techniques, e.g., using splines
- In practice, "cheap" interpolation along with a smoothing filter is employed.

Digital-to-Analog Conversion



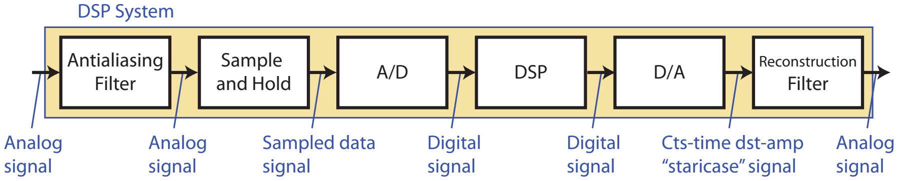
- Common interpolation approaches: bandlimited interpolation, zero-order hold, linear interpolation, higher-order interpolation techniques, e.g., using splines
- In practice, "cheap" interpolation along with a smoothing filter is employed.

Digital-to-Analog Conversion



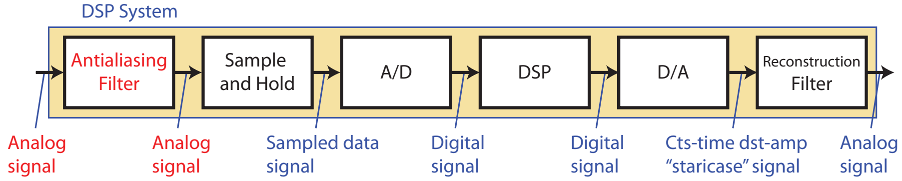
- Common interpolation approaches: bandlimited interpolation, zero-order hold, linear interpolation, higher-order interpolation techniques, e.g., using splines
- In practice, "cheap" interpolation along with a smoothing filter is employed.

A DSP System



- In practice, a DSP system does not use idealized A/D or D/A models.

A DSP System

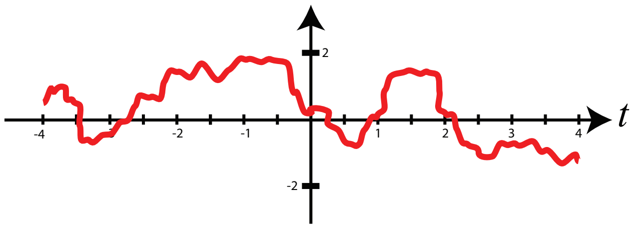


Anti-aliasing Filter:

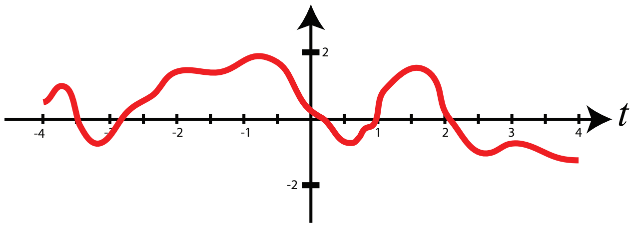
- ensures that analog input signal does not contain frequency components higher than half of the sampling frequency (to obey the sampling theorem)
- this process is irreversible

A DSP System

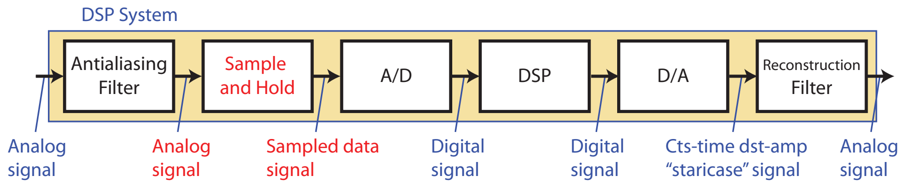
Input Signal



Anti-aliased Signal



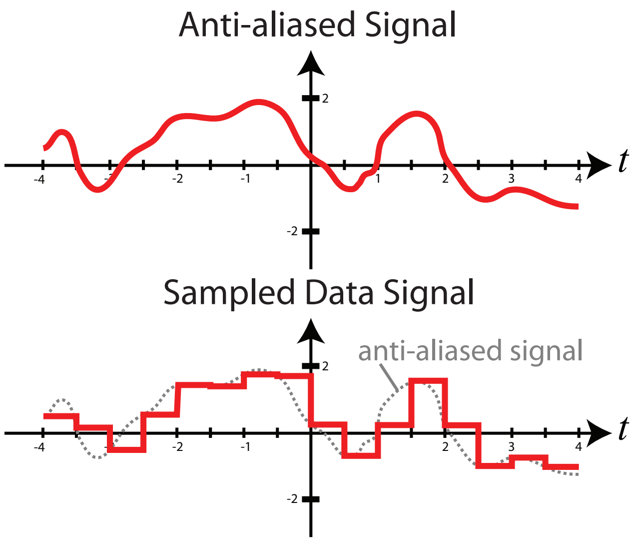
A DSP System



Sample and Hold:

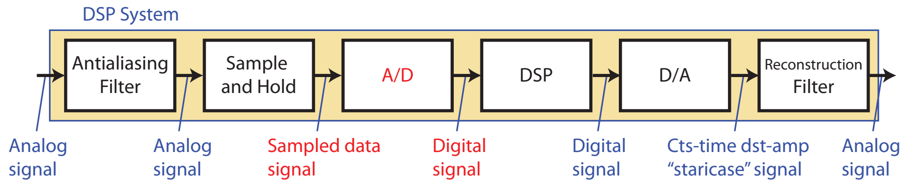
- holds a sampled analog value for a short time while the A/D converts and interprets the value as a digital

A DSP System



DSP Systems

A DSP System

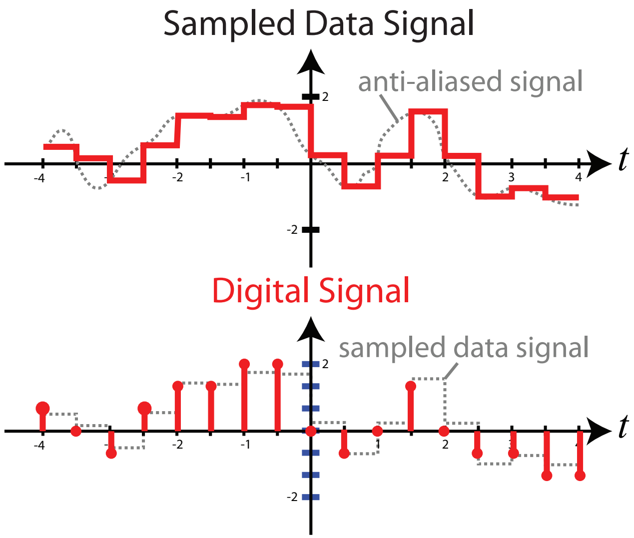


A/D:

- converts a sampled data signal value into a digital number, in part, through quantization of the amplitude

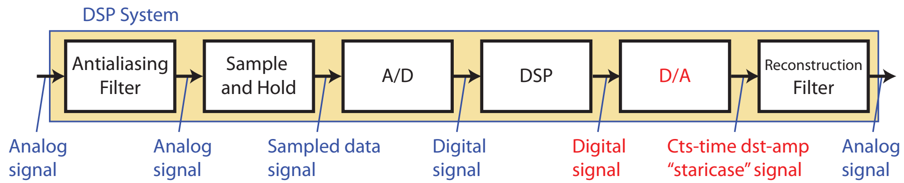
DSP Systems

A DSP System



DSP Systems

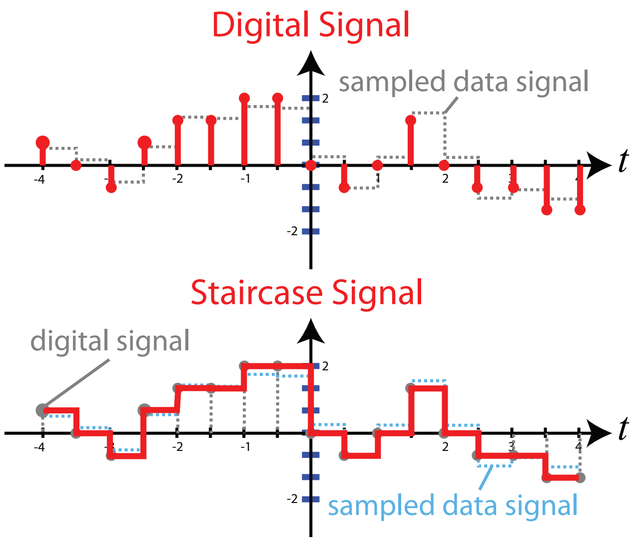
A DSP System



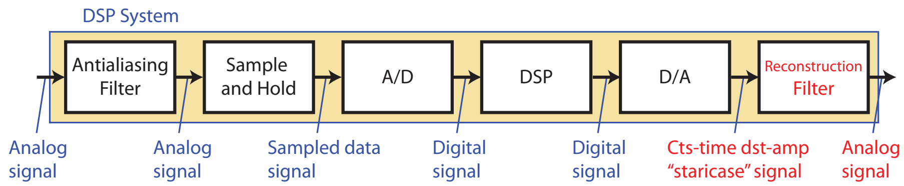
D/A:

- converts a digital signal into a "staircase"-like signal

A DSP System



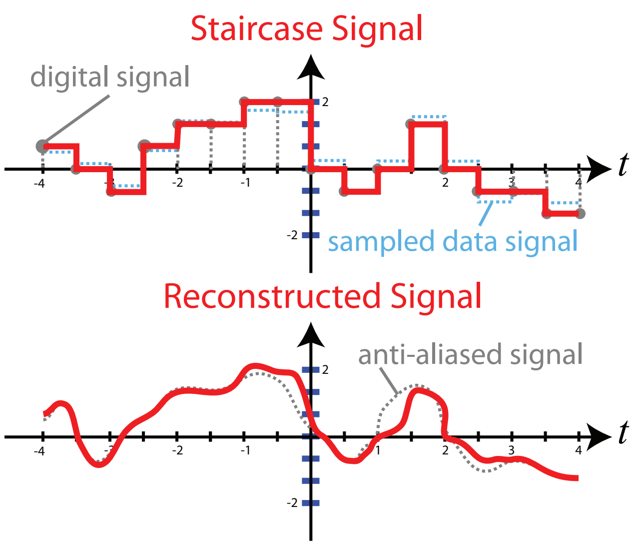
A DSP System



Reconstruction Filter:

- converts a "staircase"-like signal into an analog signal through lowpass filtering similar to the type used for anti-aliasing

A DSP System



Real-time DSP Considerations

Q: What are initial considerations when designing a DSP system that must run in real-time?

- Algorithm: related to computational operations and accuracy required by the application
- Sample rate: the rate at which input samples are received for processing
- Speed: to meet an application throughput requirement with a given sample rate, it must be possible to operate the DSP at a particular speed
- Numeric representation: format and number of bits used for data representation; depends on required computational precision and dynamic range required for application

Real-time DSP Considerations

Q: Is a DSP technology suitable for a real-time application?

- Clock rate: rate at which a DSP performs its most basic unit of work; to meet the timing requirement with a given sampling rate, it must be possible to operate the DSP at a particular clock rate
- Throughput: rate of multiply and accumulates (MACs) performed; measured in number of MACs per second
- Arithmetic and addressing capability: requirements related to the algorithm complexity, precision and data access
- Precision: associated with format (fixed vs. floating), number of bits used for data representation, and required dynamic range
- Size, cost and power consumption: technology-dependent

Programmable DSPs

- Application-specific: designed to perform one function more accurately, faster or more cost-effectively
  - examples: FFT chips, digital filters
  - can be programmable within confines of a function; e.g., coefficients of a digital filter
- General purpose: microprocessor whose architecture is optimized to process sampled data at high rates via pipelining and parallelism
  - programmable and more cost-effective for general computing
  - short system design cycle time

■